An introduction to the `rfishbase` Package

Carl Boettiger[1], Peter Wainwright[1]

[a] *Center for Population Biology, University of California, Davis, United States*

Abstract

We introduce a package that provides programmatic access to the FishBase repository~[**?**]. This package allows us to interact with data on over 30,000 fish species in the rich statistical computing environment, `R`, and can also allow the user to integration with other research software, such as the phylogenetics package~`ape`.

Keywords: R, vignette, fishbase

1. Introduction

Describe the fishbase database ... extent, information avialable, etc Briefly describe R
Stuff about Machine access to data, role of largescale data in ecology [**?**], [**?**], [**?**]
?]

2. Examples

The `rfishbase` package works by creating a cached copy of all data on fishbase currently available in XML format. Caching increases the speed of queries and places minimal demands on the fishbase server, which in it's present form is not built to support direct access to application programming interfaces (APIs). The cached copy can be loaded in to R using the command:

```
## Loading required package: rfishbase
```

```
data(fishbase)
```

To get the most recent copy of fishbase, update the cache instead. The update may take up to 24 hours. This copy is stored in the working directory with the current date and can be loaded when finished.

```
updateCache()
loadCache("2011-10-12fishdata.Rdat")
```

Loading the database creates an object called fish.data, with one entry per fish species for which data was successfully found, for a total of

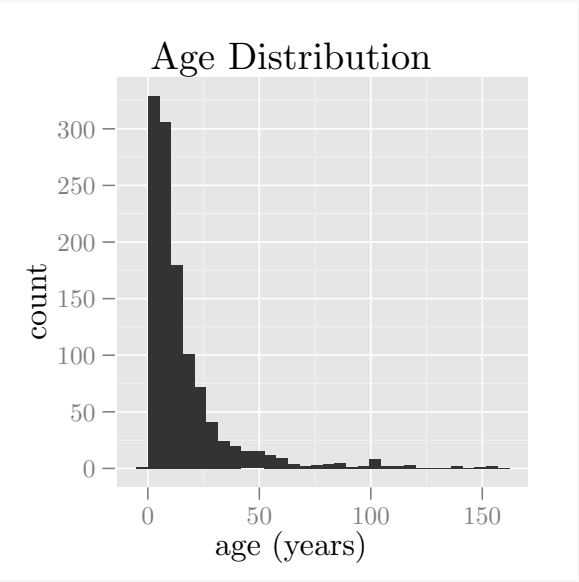
```
length(fish.data)

## [1] 30622
```

2.1. Simple queries

The examples we give here are meant to be illustrative of the kinds of queries that are possible, and also provide a simple introduction to assist the reader in using the software itself.

*Corresponding author.

```
yr <- getSize(fish.data, "age")
nfish <- length(fish.data)
qplot(yr, main = "Age Distribution", xlab = "age (years)")

## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



Typical use of the package constructs queries to identify those species matching certain criteria, which can easily be combined with other queries to quickly answer potentially questions that would otherwise be tedious to evaluate. For instance, we can ask "are there more labrids or goby species of reef fish?" using the following queries:

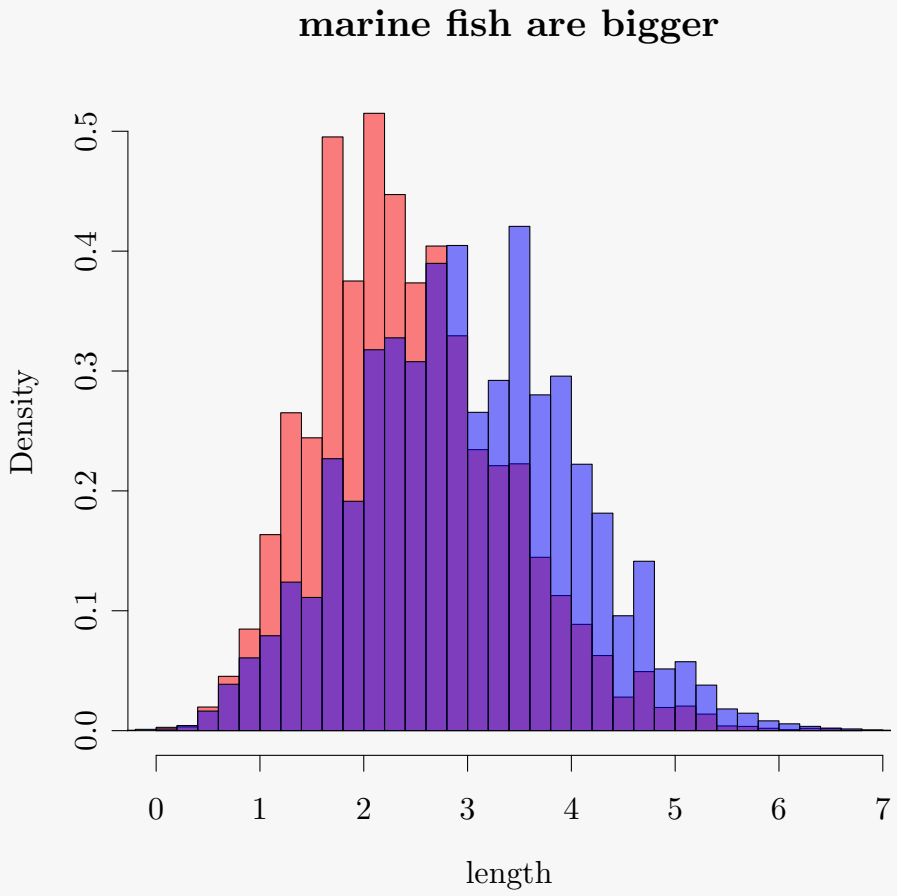
```
# Labrids include parrotfish, Scaridae, indicated by
#   the 'or' symbol |
labrid <- familySearch("(Labridae|Scaridae)",
    fish.data)
goby <- familySearch("Gobiidae", fish.data)
# get all the labrids that are reefs:
labrid.reef <- habitatSearch("reef", fish.data[labrid])
# How many species are reef labrids:
sum(labrid.reef)  # same as: length(fish.data[labrid][labrid.reef])

## [1] 503

# How many reef gobies:
sum(habitatSearch("reef", fish.data[goby]))

## [1] 401
```

Note that any function can take a subset of the data, as specified by the square brackets.

2.2. Interfacing with other software

One of the greatest advantages about accessing fishbase directly through R is the ability to take advantage of the suite of specialized analyses available through R packages. Likewise, users familiar with these packages can more easily take advantage of the data available on fishbase. We illustrate this with an example that combines phylogenetic methods available in R with quantitative trait data available from `rfishbase`.

```
# Let's plot the log length distribution of freshwater
#   vs marine fish:
hist(log(getSize(fish.data[habitatSearch("freshwater",
    fish.data)], "length")), col = rgb(1, 0, 0, 0.5), breaks = 40,
    freq = F, xlab = "length", main = "marine fish are bigger")
hist(log(getSize(fish.data[habitatSearch("marine",
    fish.data)], "length")), col = rgb(0, 0, 1, 0.5), breaks = 40,
    add = T, freq = F)
```



This series of commands illustrates testing for a phylogenetically corrected correlation between the maximum observed size of a species and the maximum observed depth at which it is found.

```
# load a phylogenetic tree
data(labridtree)

# Find those species on FishBase
tree$tip.label <- gsub("_", " ", tree$tip.label)
tip.labels <- tree$tip.label
myfish <- findSpecies(tip.labels, fish.data)
species.names <- sapply(fish.data[myfish], function(x) x$ScientificName)


# Get the maxium depth of each species
depths <- getDepth(fish.data[myfish])
rownames(depths) <- species.names

# also get the maximum size of each species
size <- getSize(fish.data[myfish], "length")
names(size) <- species.names

# Drop unmatched or missing data
missing <- tip.labels[!tip.labels %in% species.names]
tr <- drop.tip(tree, missing)
depth <- depths[, 2]
missing <- names(depth[is.na(depth)])
tr <- drop.tip(tr, missing)
missing <- names(size[is.na(size)])
tr <- drop.tip(tr, missing)

# drop all the data not in the tree
pruned.names <- tr$tip.label
size <- size[pruned.names]
depth <- depth[pruned.names]

# Does depth correlate with size after correcting for
#   phylogeny?
x <- pic(size, tr)
y <- pic(depth, tr)
summary(lm(y ~ x - 1))  # Yes

##
## Call:
## lm(formula = y ~ x - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.397  -3.488  -0.794   2.021  17.156
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   0.1686     0.0779    2.16    0.033 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 5.49 on 98 degrees of freedom
## Multiple R-squared: 0.0456,Adjusted R-squared: 0.0359
## F-statistic: 4.68 on 1 and 98 DF,  p-value: 0.0329
##
```

We can also estimate different evolutionary models for these traits to decide which best describes the data.

```
bm <- fitContinuous(tr, depth, model = "BM")[[1]]

## Fitting  BM model:

ou <- fitContinuous(tr, depth, model = "OU")[[1]]

## Fitting  OU model:

bm$aic < ou$aic

## [1] FALSE
```

In a similar fashion, programmers of other R software packages can make use of the rfishbase package to make this data available to their functions, further increasing the use and impact of fishbase. For instance, the project OpenFisheries.org makes use of the fishbase package to provide information about commercially relevant species.

3. Discussion

3.1. The self-updating study

Describe how this package could help make studies that could be automatically updated as the dataset is improved and expanded (like the examples in this document which are automatically run when the pdf is created). [**?**], [**?**].

3.2. Limitations and future directions

Fishbase contains much additional data that has not been made accessible in it's machine-readable XML format. We are in contact with the database managers and look forward to providing access to additional types of data as they become available.

4. Acknowledgements

CB is supported by a Computational Sciences Graduate Fellowship from the Department of Energy under grant number DE-FG02-97ER25308.